

Docket No.: 003829.P007  
Express Mail No.: EL651820213US

**UNITED STATES PATENT APPLICATION**

**FOR**

**METHOD FOR CONFIGURING A NETWORK ELEMENT  
HAVING AN UNKNOWN PROTOCOL ADDRESS**

Inventor:

Kim F. Storm

Prepared By:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN  
12400 Wilshire Blvd., 7th Floor  
Los Angeles, California 90025-1026  
(310) 207-3800

# METHOD FOR CONFIGURING A NETWORK ELEMENT HAVING AN UNKNOWN PROTOCOL ADDRESS

## BACKGROUND

### Field of the Invention

[0001] The invention relates to networking. More specifically, the invention relates to configuration of a network element having an unknown protocol address.

### Background

[0002] Internet protocol (IP) is a predominant networking protocol in use today. This is due at least in part to the fact that IP divides the network into subnets and is therefore highly scaleable and suitable for implementation of very large networks. Using IP, the management device is only able to access and therefore configure those devices having a same subnet address as the management device. Accordingly, for configuration purposes, a network element with an unknown IP address typically would be coupled to the management device via a serial port. The network element could then be manually configured via the serial port to use an IP address in the same subnet of the management device. Without a serial port, configuring a network element without a known IP address would normally require an external reset of the network element's configuration (typically via a reset switch) to force the network element to use a well-defined IP address, and then reconfigure the management device to use an IP address in the same subnet. When the number of devices to be configured increases, the inconvenience of this method of configuration similarly increases.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

[0003] The invention is illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to "an" or "one" embodiment in this disclosure are not necessarily to the same embodiment, and such references mean at least one.

[0004] **Figure 1** is a block diagram of the system of one embodiment of the invention.

[0005] **Figure 2A** is a schematic diagram of a typical prior art Ethernet frame.

[0006] **Figure 2B** is a schematic diagram of Ethernet frame that might be constructed by a management node in one embodiment of the invention.

[0007] **Figure 3** is a flow diagram of operation by the management node in one embodiment of the invention.

[0008] **Figure 4** is a flow diagram of an operation of a network element in one embodiment of the invention.

## **DETAILED DESCRIPTION**

[0009] **Figure 1** is a block diagram of the system of one embodiment of the invention. Ethernet 100 couples a management node 102 to a user node 106 and the device A 104 and device B 108. Device A 104 and device B 108 are network elements. In one embodiment, the network element may be for example, the InterJak™ 200 available from Filanet Corporation of Sunnyvale, California. As described, Ethernet 100 forms a physical subnet of management node 102. As used herein, "physical subnet" is deemed to include all network elements on the local network reachable without passing through a router.

[0010] Device A 104 includes a direct internet protocol (DIP) module 120 which is described in more detail below. Device A 104 is able to receive and process packets directly to the DIP module 120. Device B 108 is not able to do this, so it includes a packet filter 126 to snoop lower layers of the protocol stack B 119 and copy the relevant packets to the DIP module 120. Device A 104 also has an external port 124 by which it is coupled to internet 110. Device A 104 also has a local port 122 by which is connected to management node 102 over the Ethernet 100. In one embodiment, the DIP module is only active on local port 122 and the DIP module may also only be enabled for a limited time after power up. Such an embodiment reduces the risk of an intentional or unintentional interruption in connectivity resulting from reconfiguration.

[0011] At power up, it is presumed that the management node 102 does not know the IP address for device A 104 or device B 108. Moreover, management node 102 has no assurance that such devices even have a same subnet address as the management node 102. Thus, under traditional IP it may not be possible for the management node to interact with and/or configure device A 104 or device B 108. However, when the DIP module 120 is active, management node 102 may create a broadcast frame appropriately directed to be received by the DIP module 120 and broadcast it over the physical subnet. In its simplest form, the physical subnet could be merely the management node

coupled directly by an Ethernet cable to a single device. However, a more expansive subnets such as that shown in Figure 1 are within the scope and contemplation of the invention.

[0012] Device A 104 and device B 108 will each respond to the broadcast packet, thereby providing the management node 102 with their current respective addresses. If each such address is within the management node's subnet, the management node is able to immediately connect via those addresses and appropriately configure the network elements. However, when the subnet addresses do not match, the management node 102 may force either device A 104 or device B 108 to change its IP address to one within the management node's subnet. In doing this, the management node 102 must identify an unused IP address within its subnet and provide that address to only one of the network elements. Details of this operation will be described in referenced Figure 3 below. The other device may subsequently be forced to change its address as well by repeating the procedure.

[0013] **Figure 2A** is a schematic diagram of a typical prior art Ethernet frame. A typical Ethernet frame includes a series of headers. A hardware header includes a hardware address field and a protocol specification field in this example specifying IP protocol. An IP header includes an IP address field and protocol field, in this case, specifying transmission control protocol (TCP). The IP header is followed by a TCP header specifying a TCP port indicating an application protocol, in this case, hypertext transfer protocol (HTTP). Next comes an HTTP header with an HTTP request code. This generalized format is common to existing Ethernet frames.

[0014] **Figure 2B** is a schematic diagram of Ethernet frame that might be constructed by a management node in one embodiment of the invention. A hardware header 200 includes a hardware address field 210 and a hardware protocol field 212, e.g., specifying the IP protocol. The protocol header 202 includes a protocol address field 214 and a subprotocol field 216, e.g, specifying user datagram protocol (UDP). Subprotocol header 204 includes a port field 218

specifying an application protocol, direct IP in this case. A direct IP header is also provided, including a direct IP request code field 222. The direct IP header may also include a hardware address field to identify the target network element and a field to contain the network elements' IP address in responses or a forced IP address in force requests. In some embodiments, control or status fields are also provided. In one such embodiment, a flag is set in a defined status field when the time during which the DIP module is active, e.g., the address can be forced, has expired. By appropriately setting the hardware address field 210 and the protocol address field 214 to indicate all addresses and by selecting UDP as the subprotocol in the subprotocol field 216, the frame will not be screened out by the protocol stack in devices on the physical subnet to be configured even when the devices have a different subnet address. By appropriately selecting the UDP port number to be one recognized by the DIP module, if that module is enabled, it will accept the frame and handle the frame appropriately. Unlike TCP, UDP is not connection based and is therefore more suitable for a generalized case of an unknown device address.

[0015] Alternatively, the device may be provided with a packet filter that permits the device to snoop at the hardware level and then require the protocol stack 119 to, e.g., only screen based on hardware address and port number. In this manner, regardless of the IP header, if the hardware address and port number are directed to the DIP module, the frame will be forwarded to the DIP module. Devices operating with a linux kernel support this packet filtering capability as well as the other embodiment described above.

[0016] **Figure 3** is a flow diagram of operation by the management node in one embodiment of the invention. At functional block 302, the management node sends a broadcast frame with a defined port number (consistent with direct IP) to all devices on the physical subnet. Because any devices not having a DIP module will discard the packet, if a response is received, a device exists that may need configuration. At decision block 304, the response is checked for the respondents current protocol, e.g., (IP) address at functional block 306. At

decision block 308, a determination is made if the current protocol address is in the access range of the management node. For example, is the protocol address within the same subnet? If the current protocol address is not within the same subnet (access range), the management node iteratively queries addresses within the subnet until an unused address is found at functional block 310. In one embodiment, the management node may use internet control management protocol (ICMP) echo request (commonly known as a "ping") to perform the query. As this is a standard way to determine connectivity, the absence of a response to a ping indicates an unused IP address. In another embodiment, the management node may use the address resolution protocol (ARP) to perform the query. At functional block 312 of the management node creates and sends a frame to the respondent to force the respondent to change his address to the unused address identified in functional block 310. In one embodiment, this forcing frame may be constructed as a broadcast frame with a hardware address designated as "All," an IP address as "All," the appropriate UDP port, and the target hardware address in a direct IP header field. Alternatively, in another embodiment, the hardware address of the target device may be used in the hardware address header of the forcing packet. By using the hardware address as a target identifier, the risk of two devices on the physical subnet being forced to a single address is ameliorated. Once the change is complete or if at decision block 308 the existing address is within the management node's access range, the management node may connect normally to the respondent using the then existing address and configure the device normally over the network. As applied to IP configuration, the above method permits connection between a device and to management node using TCP/IP after the exchange of only three Ethernet frames. Notably, the configuration of the management node need not change and no reboot is required. Moreover, configuration can be accomplished in the absence of a serial port via a standard Ethernet connection.

[0017] **Figure 4** is a flow diagram of an operation of a network element in one embodiment of the invention. At functional block 402, the device receives a

broadcast packet. At decision block 404, a determination is made if the broadcast frame received is for the port and of the appropriate format for the DIP module of the device. In one embodiment packet filtering is used. In another embodiment no filtering is required because the frame is constructed to pass through the protocol stack without filtering. If it is, at functional block 406, the device builds a response frame including its hardware address and its current protocol address. In functional block 408, it sends the frame to the source of the broadcast frame. In one embodiment, the device is able to send the response at a sufficiently low level, e.g., the hardware level, that the frame can be specifically directed to the broadcast source. In another embodiment, the device creates a broadcast frame with the original broadcaster's hardware address contained in an appropriate field within the direct IP header. At functional block 410, a forcing broadcast frame is received. At decision block 412, a determination is made if the forcing frame is directed to the correct number and matches the hardware address of the device. If it does not match the port number and hardware address at decision 412 or the port and format were not okay at decision block 404, the frame is discarded at functional block 414. If the port number and hardware address are okay at decision block 412, the device changes its protocol address to the address specified in the forcing frame at functional block 416.

**[0018]** In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes can be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.